

Safety Alternating Automata on Data Words

RANKO LAZIĆ

Department of Computer Science, University of Warwick, UK

A data word is a sequence of pairs of a letter from a finite alphabet and an element from an infinite set, where the latter can only be compared for equality. Safety one-way alternating automata with one register on infinite data words are considered, their nonemptiness is shown EXPSPACE-complete, and their inclusion decidable but not primitive recursive. The same complexity bounds are obtained for satisfiability and refinement, respectively, for the safety fragment of linear temporal logic with freeze quantification. Dropping the safety restriction, adding past temporal operators, or adding one more register, each causes undecidability.

Categories and Subject Descriptors: F.4.1 [Mathematical Logic and Formal Languages]: Formal Languages—*Decision problems*; F.1.1 [Computation by Abstract Devices]: Models of Computation—*Automata*

General Terms: Algorithms, Verification

1. INTRODUCTION

Context. Logics and automata for words and trees over finite alphabets are relatively well-understood. Motivated partly by the need for formal verification and synthesis of infinite-state systems, and the search for automated reasoning techniques for XML, there is an active and broad research programme on logics and automata for words and trees which have richer structure.

Segoufin’s survey [Segoufin 2006] is a summary of the substantial progress made on reasoning about data words and data trees. A data word is a word over a finite alphabet, with an equivalence relation on word positions. Implicitly, every word position is labelled by an element (“datum”) from an infinite set (“data domain”), but since the infinite set is equipped only with the equality predicate, it suffices to know which word positions are labelled by equal data, and that is what the equivalence relation represents. Similarly, a data tree is a tree (countable, unranked and ordered) whose every node is labelled by a letter from a finite alphabet, with an equivalence relation on the set of its nodes.

It has been nontrivial to find satisfactory specification formalisms even for data words. First-order logic was considered in [Bojańczyk et al. 2006; David 2004], and related automata were studied further in [Björklund and Schwentick 2007]. The

This paper is a revised and extended version of [Lazić 2006].

This research was supported by grants from the EPSRC (GR/S52759/01) and the Intel Corporation, and by ENS Cachan.

Permission to make digital/hard copy of all or part of this material without fee for personal or classroom use provided that the copies are not made or distributed for profit or commercial advantage, the ACM copyright/server notice, the title of the publication, and its date appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or a fee.

© 20YY ACM 0000-0000/20YY/0000-0001 \$5.00

ACM Journal Name, Vol. V, No. N, Month 20YY, Pages BD.

logic has variables which range over word positions ($\{0, \dots, l-1\}$ or \mathbb{N}), a unary predicate for each letter from the finite alphabet, and a binary predicate $x \sim y$ for the equivalence relation that represents equality of data labels. $\text{FO}^2(\sim, <, +1)$ denotes such a logic with two variables and binary predicates $x + 1 = y$ and $x < y$. Over finite and over infinite data words, satisfiability for $\text{FO}^2(\sim, <, +1)$ was proved decidable and at least as hard as reachability for Petri nets [Bojańczyk et al. 2006]. The latter problem is EXPSPACE -hard [Lipton 1976], but its elementarity is still an open question. Elementary complexity of satisfiability can be obtained at the price of substantially reducing the navigational power: over finite data words, NEXPTIME -completeness for $\text{FO}^2(\sim, <)$ was established in [David 2004] and 3NEXPTIME -membership for $\text{FO}^2(\sim, +1)$ follows from [Bojańczyk et al. 2006]. In the other direction, if $\text{FO}^2(\sim, <, +1)$ is extended by one more variable, $+1$ becomes expressible using $<$, but satisfiability was shown undecidable already for $\text{FO}^3(\sim, +1)$ [Bojańczyk et al. 2006].

An alternative approach to reasoning about data words is based on automata with registers [Kaminski and Francez 1994]. A register is used for storing a datum for later equality comparisons (i.e. an equivalence class for later membership testing). Nonemptiness of one-way nondeterministic register automata over finite data words has relatively low complexity: NP -complete [Sakamoto and Ikeda 2000] or PSPACE -complete [Demri and Lazić 2009], depending on technical details of their definition. Unfortunately, such automata fail to provide a satisfactory notion of regular language of finite data words, as they are not closed under complement [Kaminski and Francez 1994] and their nonuniversality is undecidable [Neven et al. 2004]. To overcome those limitations, one-way alternating automata with 1 register (for short, 1ARA_1) were proposed in [Demri and Lazić 2009]: they are closed under Boolean operations, their nonemptiness over finite data words is decidable, and future-time fragments of temporal logics such as LTL or the modal μ -calculus extended by 1 register are easily translatable to such automata. However, nonemptiness for 1ARA_1 turned out to be not primitive recursive over finite data words, and undecidable (more precisely, Π_1^0 -hard) over infinite ones with the weak acceptance mechanism [Muller et al. 1986] and thus also with Büchi or co-Büchi acceptance.

Contribution. We consider one-way alternating automata with 1 register with the safety acceptance mechanism over infinite data words (i.e. data ω -words). The languages of such automata are safety properties [Alpern and Schneider 1987]: every rejected data ω -word has a finite prefix such that every other data ω -word which extends it is also rejected. (Over finite data words, safety is not a restriction.)

The main result is that nonemptiness of safety 1ARA_1 is in EXPSPACE . We say that a sentence of LTL is safety iff each occurrence of the ‘until’ operator is under an odd number of negations. In particular, each ‘eventually’ (resp., ‘always’) must be under an odd (resp., even) number of negations. By showing that the safety fragment of future-time LTL with 1 register is translatable in logarithmic space to safety 1ARA_1 , and that satisfiability for the fragment is EXPSPACE -hard, we conclude EXPSPACE -completeness of both problems.

The EXPSPACE upper bound is surprising since even decidability is fragile: by [Demri and Lazić 2009, Theorem 5.2], satisfiability for future-time LTL with 1 register on data ω -words is Π_1^0 -hard, and from the proof of [Demri and Lazić 2009,

Theorem 5.4], the same is true for the safety fragment if past temporal operators or one more register are added (cf. related undecidability results in [Neven et al. 2004; David 2004]). Moreover, nonemptiness of safety forward (i.e. downward and rightward) alternating automata with 1 register on data trees was shown decidable but not elementary [Jurdziński and Lazić 2007]. Another setting where decidability [Ouaknine and Worrell 2006] was obtained by restricting to safety sentences is that of metric temporal logic on timed ω -words, but the complexity is again not elementary [Bouyer et al. 2008].

The proof of EXPSpace-membership is in two stages. The first consists of translating a given safety 1ARA_1 \mathcal{A} to a nondeterministic automaton with faulty counters $\mathcal{C}_{\mathcal{A}}$ which is on ω -words over the alphabet of \mathcal{A} and which is nonempty iff \mathcal{A} is. The counters of $\mathcal{C}_{\mathcal{A}}$ are faulty in the sense that they are subject to incrementing errors, i.e. they can spontaneously increase at any time. Although a nonemptiness-preserving translation from 1ARA_1 with weak acceptance to counter automata with incrementing errors was given in [Demri and Lazić 2009], applying it to safety 1ARA_1 produces automata with the Büchi acceptance mechanism, where the latter ensures that certain loops cannot repeat infinitely due to incrementing errors. To obtain safety automata, we enrich the instruction set by *nondeterministic transfers*. When applied to a counter c and a set of counters C , such an instruction transfers the value of c to the counters in C , nondeterministically splitting it. Thus we obtain $\mathcal{C}_{\mathcal{A}}$ whose nonemptiness amounts to existence of an infinite computation from the initial state. However, a further observation on the resulting automata is required: the counters of such an automaton are nonempty subsets of a certain set (essentially, the set of states of the given safety 1ARA_1), and it suffices to use nondeterministic transfers which are simultaneous for all counters and which have a certain distributivity property in terms of the partial-order structure of the set of all counters.

The second stage of the proof is then an inductive counting argument which shows that $\mathcal{C}_{\mathcal{A}}$ is nonempty iff it has a computation from the initial state of length doubly exponential in the size of \mathcal{A} . Some of the techniques are also used in the proof that termination of channel machines with occurrence testing and insertion errors is primitive recursive [Bouyer et al. 2008]. Although counters are simpler resources than channels, the class of machines considered do not have instructions which correspond to the nondeterministic transfers, and the sets of channels and messages (which are counterparts to the sets of counters) have no special structure.

We also show that language inclusion between two safety 1ARA_1 is decidable, and hence that refinement (i.e., validity of implication) between two sentences of safety future-time LTL with 1 register is also decidable. Since the safety fragment is closed under conjunctions and disjunctions, it follows that satisfiability is decidable for Boolean combinations of safety sentences. The latter is thus a competing logic to $\text{FO}^2(\sim, <, +1)$ on data ω -words. They are incomparable in expressiveness: there exist properties involving the past (e.g. ‘every b is preceded by an a with the same datum’) which are expressible in $\text{FO}^2(\sim, <, +1)$ but not by a Boolean combination of safety sentences (not even in future-time LTL with 1 register), and the reverse is true of some constraints involving more than 2 word positions (e.g. ‘whenever a is followed by b with the same datum, c does not occur in between’). However,

as pointed out above, it is not known whether satisfiability for $\text{FO}^2(\sim, <, +1)$ is elementary, whereas we establish that already satisfiability for negations of safety sentences is not primitive recursive, and hence also universality for safety 1ARA_1 .

2. PRELIMINARIES

In this section, we define safety one-way alternating automata and safety future-time linear temporal logic with 1 register on data ω -words, as well as the class of counter automata that will be used in the proof of EXPSPACE -membership in Section 3. We also show some of their basic properties, in particular a logarithmic-space translation from the linear temporal logic to the alternating automata.

2.1 Data Words

A *data ω -word* σ over a finite alphabet Σ is an ω -word $\text{str}(\sigma)$ over Σ together with an equivalence relation \sim^σ on $\mathbb{N} = \{0, 1, \dots\}$. We write \mathbb{N}/\sim^σ for the set of all classes of \sim^σ . For $i \in \mathbb{N}$, we write $\sigma(i)$ for the letter at position i , and $[i]_{\sim^\sigma}$ for the class that contains i . When σ is understood, we may write simply \sim instead of \sim^σ . We shall sometimes refer to classes of \sim as ‘data’.

In some places, we shall also need the concept of a finite data word. For $i > 0$, the i -prefix of a data ω -word σ is the finite data word whose letters are $\sigma(0) \cdots \sigma(i-1)$ and whose equivalence relation is \sim^σ restricted to $\{0, \dots, i-1\}$.

2.2 Register Automata

The definition of safety one-way alternating 1-register automata below is based on the more general one of weak two-way alternating register automata in [Demri and Lazić 2009]. A configuration of such an automaton at a position i of a data ω -word σ will consist of one of finitely many automaton states and a register value $D \in \mathbb{N}/\sim$. From it, depending on the state, the letter $\sigma(i)$, and whether $D = [i]_{\sim}$ (denoted \uparrow) or $D \neq [i]_{\sim}$ (denoted \nmid), the automaton chooses a pair Q', Q'_\downarrow of sets of states. The resulting set of configurations at the next word position is $\{\langle q', D \rangle : q' \in Q'\} \cup \{\langle q', [i]_{\sim} \rangle : q' \in Q'_\downarrow\}$, i.e. the states in Q' are associated with the old register value, and the states in Q'_\downarrow with the class of position i . Following [Brzozowski and Leiss 1980], what choices of pairs of sets of states are possible will be specified in each case by a positive Boolean formula. That formalisation, in contrast to listing all possible such choices, will enable a logarithmic-space translation from safety future-time LTL with 1 register.

An infinite run of the automaton will consist, for each $j \in \mathbb{N}$, of a set F_j of all configurations at position j . For each j , F_{j+1} will be the union of some sets of configurations chosen as above for each configuration in F_j . Hence, a configuration will be rejecting when its set of possible choices is empty, and it will be accepting when it can choose $Q' = Q'_\downarrow = \emptyset$. The definition of infinite runs will ensure that they cannot contain rejecting configurations, so the safety acceptance mechanism will amount to each infinite run being considered accepting.

Formally, for a finite set Q , let $\downarrow Q = \{\downarrow q : q \in Q\}$, and let $\mathcal{B}_\downarrow^+(Q)$ denote the set of all positive Boolean formulae over $Q \cup \downarrow Q$, where we assume that Q and $\downarrow Q$ are disjoint:

$$\varphi ::= q \mid \downarrow q \mid \top \mid \perp \mid \varphi \wedge \varphi \mid \varphi \vee \varphi$$

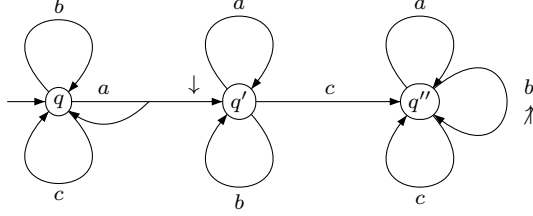


Fig. 1. A register automaton

A *safety one-way alternating automaton with 1 register* (shortly, *safety 1ARA₁*) \mathcal{A} is a tuple $\langle \Sigma, Q, q_I, \delta \rangle$ such that:

- Σ is a finite alphabet;
- Q is a finite set of states, and $q_I \in Q$ is the initial state;
- $\delta : (Q \times \Sigma \times \{\uparrow, \downarrow\}) \rightarrow \mathcal{B}_\downarrow^+(Q)$ is a transition function.

Satisfaction of a positive Boolean formula over $Q \cup \downarrow Q$ by a pair of sets $Q', Q'_\downarrow \subseteq Q$ is defined by structural recursion:

$$\begin{aligned} Q', Q'_\downarrow \models q &\stackrel{\text{def}}{\iff} q \in Q' & Q', Q'_\downarrow \models \top \\ Q', Q'_\downarrow \models \downarrow q &\stackrel{\text{def}}{\iff} q \in Q'_\downarrow & Q', Q'_\downarrow \models \perp \\ Q', Q'_\downarrow \models \varphi \wedge \varphi' &\stackrel{\text{def}}{\iff} Q', Q'_\downarrow \models \varphi \text{ and } Q', Q'_\downarrow \models \varphi' \\ Q', Q'_\downarrow \models \varphi \vee \varphi' &\stackrel{\text{def}}{\iff} Q', Q'_\downarrow \models \varphi \text{ or } Q', Q'_\downarrow \models \varphi' \end{aligned}$$

A configuration of \mathcal{A} for a data word σ is an element of $Q \times (\{j : 0 \leq j < |\sigma| \} / \sim)$. For a position $0 \leq i < |\sigma|$, and finite sets F and F' of configurations, we write $F \xrightarrow{\sigma, i} F'$ iff, for each $\langle q, D \rangle \in F$, there exist $Q^{(q, D)}, Q_\downarrow^{(q, D)} \subseteq Q$ which satisfy the formula $\delta(q, \sigma(i), \uparrow)$ if $D = [i]_\sim$, or the formula $\delta(q, \sigma(i), \downarrow)$ if $D \neq [i]_\sim$, such that

$$F' = \{ \langle q', D \rangle : \langle q, D \rangle \in F \wedge q' \in Q^{(q, D)} \} \cup \{ \langle q', [i]_\sim \rangle : \langle q, D \rangle \in F \wedge q' \in Q_\downarrow^{(q, D)} \}$$

We say that \mathcal{A} accepts a data ω -word σ over Σ iff it has an infinite run $F_0 \xrightarrow{\sigma, 0} F_1 \xrightarrow{\sigma, 1} \dots$ where $F_0 = \{ \langle q_I, [0]_\sim \rangle \}$ consists of the initial configuration. We write $L(\mathcal{A})$ for the language of \mathcal{A} , i.e. the set of all data ω -words over Σ that \mathcal{A} accepts.

Example 2.1. A safety 1ARA₁ with alphabet $\{a, b, c\}$ and three states is depicted in Figure 1. It rejects a data ω -word iff there is an occurrence of a , a subsequent occurrence of b with the same datum, and an occurrence of c between them.

The automaton is deterministic, except for the universal branching from state q at letter a . When behaviour does not depend on whether the class in the register equals the class of the current position, the two cases are not shown separately. In particular, we have $\delta(q, a, \uparrow) = \delta(q, a, \downarrow) = q \wedge \downarrow q'$. The absence of a transition from q'' labelled by b and \uparrow means that we have rejection in that case, i.e. $\delta(q'', b, \uparrow) = \perp$.

A set L of data ω -words over an alphabet Σ is called *safety* [Alpern and Schneider 1987] iff it is closed under limits of finite prefixes, i.e. for each data ω -word σ , if for each $i > 0$ there exists $\sigma'_i \in L$ with the i -prefixes of σ and σ'_i equal, then $\sigma \in L$.¹

¹Hence, a set is safety iff it is closed with respect to the Cantor metric, where the distance between two words is inversely proportional to the length of their longest common prefix.

PROPOSITION 2.2. *The language of each safety 1ARA₁ is safety.*

PROOF. Suppose that \mathcal{A} is a safety 1ARA₁, and for each $i > 0$ there exists $\sigma'_i \in L(\mathcal{A})$ such that the i -prefixes of σ and σ'_i are equal. For each i , let $F'_{i,0} \xrightarrow{\sigma'_{i,0}} F'_{i,1} \xrightarrow{\sigma'_{i,1}} \dots$ be an infinite run of \mathcal{A} with $F'_{i,0} = \{\langle q_I, [0]_{\sim \sigma'_i} \rangle\}$. For each $0 \leq j \leq i$, let $F'_{i,j}^\dagger$ be obtained from $F'_{i,j}$ by replacing each class D' of σ'_i with the class D of σ such that $D' \cap \{0, \dots, i-1\} = D \cap \{0, \dots, i-1\}$. Now, consider the tree formed by all the sequences $\langle F'_{i,j}^\dagger : 0 \leq j \leq i \rangle$ for $i > 0$. The tree is finitely branching, so by König's Lemma, it contains an infinite path $\langle F_j : j \in \mathbb{N} \rangle$. It remains to observe that $F_0 \xrightarrow{\sigma,0} F_1 \xrightarrow{\sigma,1} \dots$ and $F_0 = \{\langle q_I, [0]_{\sim \sigma} \rangle\}$. \square

Given safety 1ARA₁ \mathcal{A}_1 and \mathcal{A}_2 with alphabet Σ , it is easy to construct an automaton which recognises $L(\mathcal{A}_1) \cap L(\mathcal{A}_2)$ (resp., $L(\mathcal{A}_1) \cup L(\mathcal{A}_2)$). It suffices to form a disjoint union of \mathcal{A}_1 and \mathcal{A}_2 , and add a new initial state q_I such that $\delta(q_I, a, ?) = \delta(q_I^1, a, ?) \wedge \delta(q_I^2, a, ?)$ (resp., $\delta(q_I, a, ?) = \delta(q_I^1, a, ?) \vee \delta(q_I^2, a, ?)$) for each $a \in \Sigma$ and $? \in \{\uparrow, \nmid\}$, where q_I^1 and q_I^2 are the initial states of \mathcal{A}_1 and \mathcal{A}_2 . We thus obtain:

PROPOSITION 2.3. *Safety 1ARA₁ are closed under finite intersections and finite unions, in logarithmic space.*

2.3 Linear Temporal Logic

Safety $\text{LTL}_1^\downarrow(\mathbf{X}, \mathbf{R})$ will denote the safety fragment of future-time linear temporal logic with 1 register, whose syntax is given below. Each formula is over a finite alphabet Σ , over which the atomic formulae a range. By restricting ourselves to formulae in negation normal form, the safety restriction amounts to the ‘release’ temporal operator being available instead of its dual ‘until’. The formulae may also contain the ‘next’ temporal operator. A freeze quantification $\downarrow\phi$ binds each free occurrence of \uparrow in ϕ . Such an occurrence will evaluate to true iff the word position at the time of the freeze quantification and the word position when the occurrence of \uparrow is evaluated are in the same class.

$$\phi ::= a \mid \top \mid \perp \mid \phi \wedge \phi \mid \phi \vee \phi \mid \mathbf{X}\phi \mid \phi \mathbf{R}\phi \mid \downarrow\phi \mid \uparrow \mid \nmid$$

The ‘always’ temporal operator can be introduced by regarding $\mathbf{G}\phi$ as an abbreviation for $\perp \mathbf{R}\phi$.

For a data ω -word σ over a finite alphabet Σ , a position $i \in \mathbb{N}$, a register value $D \in \mathbb{N}/\sim$, and a formula ϕ over Σ , writing $\sigma, i \models_D \phi$ will mean that ϕ is satisfied by σ at i with respect to D . The satisfaction relation is defined as follows, where we omit the Boolean cases.

$$\begin{aligned} \sigma, i \models_D a &\stackrel{\text{def}}{\iff} \sigma(i) = a \\ \sigma, i \models_D \mathbf{X}\phi &\stackrel{\text{def}}{\iff} \sigma, i+1 \models_D \phi \\ \sigma, i \models_D \phi \mathbf{R}\psi &\stackrel{\text{def}}{\iff} \text{either for all } k \geq i, \sigma, k \models_D \psi, \text{ or for some } j \geq i, \\ &\quad \sigma, j \models_D \phi \text{ and for all } k \in \{i, \dots, j\}, \sigma, k \models_D \psi \\ \sigma, i \models_D \downarrow\phi &\stackrel{\text{def}}{\iff} \sigma, i \models_{[i]_{\sim}} \phi \\ \sigma, i \models_D \uparrow &\stackrel{\text{def}}{\iff} i \in D \end{aligned}$$

$$\sigma, i \models_D \neg \stackrel{\text{def}}{\iff} i \notin D$$

If ϕ is a sentence, i.e. contains no free occurrence of \uparrow , we may omit D since it is irrelevant and write $\sigma, i \models \phi$. Let $L(\phi)$ denote the language of ϕ , i.e. the set of all data ω -words over Σ such that $\sigma, 0 \models \phi$.

Example 2.4. Consider the following sentence ϕ over alphabet $\{a, b, c\}$:

$$\mathbf{G}(b \vee c \vee \downarrow \mathbf{XG}(a \vee b \vee \mathbf{XG}(a \vee c \vee \neg)))$$

We have $\sigma, 0 \models \phi$ iff, for each occurrence of a in σ and each later occurrence of c , there is no later still occurrence of b with the same datum as the occurrence of a , i.e. iff σ is accepted by the automaton in Example 2.1.

THEOREM 2.5. *For each sentence ϕ of safety $LTL_1^\downarrow(\mathbf{X}, \mathbf{R})$, a safety $1ARA_1 \mathcal{A}_\phi$ with the same alphabet and $L(\phi) = L(\mathcal{A}_\phi)$ is computable in logarithmic space.*

PROOF. The translation is a straightforward adaptation of the classical one from LTL to alternating automata (cf. e.g. [Vardi 1996]).

To define \mathcal{A}_ϕ with alphabet Σ of ϕ , let the set of states Q consist of all $q_{\phi'}$ such that ϕ' is either ϕ , or ψ for a subformula $\mathbf{X}\psi$ of ϕ , or a subformula $\psi\mathbf{R}\chi$ of ϕ . Let the initial state be q_ϕ . The transition function is obtained by restricting to Q the function defined below by structural recursion over the set of all $q_{\phi'}$ where ϕ' is a subformula of ϕ . The dual cases are omitted, and $?$ ranges over $\{\uparrow, \neg\}$. In the formula for $\downarrow\psi$, each occurrence of a state q' without \downarrow is substituted by $\downarrow q'$.

$$\begin{aligned} \delta(q_a, a, ?) &\stackrel{\text{def}}{=} \top & \delta(q_{\psi \wedge \chi}, a, ?) &\stackrel{\text{def}}{=} \delta(q_\psi, a, ?) \wedge \delta(q_\chi, a, ?) \\ \delta(q_a, a', ?) &\stackrel{\text{def}}{=} \perp, \text{ for } a' \neq a & \delta(q_{\mathbf{X}\psi}, a, ?) &\stackrel{\text{def}}{=} q_\psi \\ \delta(q_\top, a, ?) &\stackrel{\text{def}}{=} \top & \delta(q_{\psi \mathbf{R} \chi}, a, ?) &\stackrel{\text{def}}{=} \delta(q_\chi, a, ?) \wedge (\delta(q_\psi, a, ?) \vee q_{\psi \mathbf{R} \chi}) \\ \delta(q_\uparrow, a, \uparrow) &\stackrel{\text{def}}{=} \top & \delta(q_{\downarrow\psi}, a, ?) &\stackrel{\text{def}}{=} \delta(q_\psi, a, \uparrow)[\downarrow q' / q' : q' \in Q] \\ \delta(q_\uparrow, a, \neg) &\stackrel{\text{def}}{=} \perp \end{aligned}$$

That \mathcal{A}_ϕ is computable in logarithmic space follows by observing that, for each subformula ϕ' of ϕ , $a \in \Sigma$, and $? \in \{\uparrow, \neg\}$, a single traversal of ϕ' suffices for computing $\delta(q_{\phi'}, a, ?)$.

Equality of the languages of ϕ and \mathcal{A}_ϕ is implied by the following claim: for each subformula ϕ' of ϕ , data ω -word σ over Σ , position $i \in \mathbb{N}$, and register value $D \in \mathbb{N} / \sim$, we have $\sigma, i \models_D \phi'$ iff, for some $Q', Q'_\downarrow \subseteq Q$ such that $Q', Q'_\downarrow \models \delta(q_{\phi'}, \sigma(i), \uparrow)$ if $D = [i]_\sim$, or such that $Q', Q'_\downarrow \models \delta(q_{\phi'}, \sigma(i), \neg)$ if $D \neq [i]_\sim$, \mathcal{A}_ϕ has an infinite run from position $i + 1$ of σ , starting with

$$\{\langle q', D \rangle : q' \in Q'\} \cup \{\langle q', [i]_\sim \rangle : q' \in Q'_\downarrow\}$$

(If $q_{\phi'}$ is a state of \mathcal{A}_ϕ , the latter is equivalent to \mathcal{A}_ϕ having a run from position i of σ , starting with $\{\langle q_{\phi'}, D \rangle\}$.) The claim is provable by structural induction on ϕ' . We treat explicitly the two interesting cases: $\phi' = \psi\mathbf{R}\chi$ and $\phi' = \downarrow\psi$.

Suppose $\sigma, i \models_D \psi\mathbf{R}\chi$. If $\sigma, j \models_D \psi$ for some $j \geq i$, and $\sigma, k \models_D \chi$ for all $k \in \{i, \dots, j\}$, then by the inductive hypothesis:

- (i) for some $Q', Q'_\downarrow \subseteq Q$ such that $Q', Q'_\downarrow \models \delta(q_\psi, \sigma(j), \uparrow)$ if $D = [j]_\sim$, or such that $Q', Q'_\downarrow \models \delta(q_\psi, \sigma(j), \neg)$ if $D \neq [j]_\sim$, \mathcal{A}_ϕ has an infinite run $F'_{j+1} \xrightarrow{\sigma, j+1} F'_{j+2} \xrightarrow{\sigma, j+2}$

... with

$$F'_{j+1} = \{\langle q', D \rangle : q' \in Q'\} \cup \{\langle q', [j]_{\sim} \rangle : q' \in Q'_{\downarrow}\}$$

- (ii) for all $k \in \{i, \dots, j\}$, for some $Q^k, Q^k_{\downarrow} \subseteq Q$ such that $Q^k, Q^k_{\downarrow} \models \delta(q_{\chi}, \sigma(k), \uparrow)$ if $D = [k]_{\sim}$, or such that $Q^k, Q^k_{\downarrow} \models \delta(q_{\chi}, \sigma(k), \nearrow)$ if $D \neq [k]_{\sim}$, \mathcal{A}_{ϕ} has an infinite run $F^k_{k+1} \xrightarrow{\sigma, k+1} F^k_{k+2} \xrightarrow{\sigma, k+2} \dots$ with

$$F^k_{k+1} = \{\langle q', D \rangle : q' \in Q^k\} \cup \{\langle q', [k]_{\sim} \rangle : q' \in Q^k_{\downarrow}\}$$

Letting $F_l^{\dagger} = \{\langle q_{\psi R_{\chi}}, D \rangle\} \cup \bigcup_{k \in \{i, \dots, l-1\}} F_l^k$ for each $l \in \{i, \dots, j\}$, and $F_l^{\dagger} = \bigcup_{k \in \{i, \dots, j\}} F_l^k \cup F_l'$ for each $l \geq j+1$, we have by (i) and (ii) that $F_i^{\dagger} \xrightarrow{\sigma, i} F_{i+1}^{\dagger} \xrightarrow{\sigma, i+1} \dots$ and $F_i^{\dagger} = \{\langle q_{\psi R_{\chi}}, D \rangle\}$, as required. If $\sigma, k \models_D \chi$ for all $k \geq i$, the argument is simpler.

For the converse, suppose \mathcal{A}_{ϕ} has an infinite run $F_i^{\dagger} \xrightarrow{\sigma, i} F_{i+1}^{\dagger} \xrightarrow{\sigma, i+1} \dots$ with $F_i^{\dagger} = \{\langle q_{\psi R_{\chi}}, D \rangle\}$. If there exists $j \geq i$ with $\langle q_{\psi R_{\chi}}, D \rangle \notin F_{j+1}^{\dagger}$, consider the minimum such j . Since $\delta(q_{\psi R_{\chi}}, a, ?) = \delta(q_{\chi}, a, ?) \wedge (\delta(q_{\psi}, a, ?) \vee q_{\psi R_{\chi}})$, we obtain:

- (iii) for some $Q', Q'_{\downarrow} \subseteq Q$ such that $Q', Q'_{\downarrow} \models \delta(q_{\psi}, \sigma(j), \uparrow)$ if $D = [j]_{\sim}$, or such that $Q', Q'_{\downarrow} \models \delta(q_{\psi}, \sigma(j), \nearrow)$ if $D \neq [j]_{\sim}$, we have

$$\{\langle q', D \rangle : q' \in Q'\} \cup \{\langle q', [j]_{\sim} \rangle : q' \in Q'_{\downarrow}\} \subseteq F_{j+1}^{\dagger}$$

- (iv) for all $k \in \{i, \dots, j\}$, for some $Q^k, Q^k_{\downarrow} \subseteq Q$ such that $Q^k, Q^k_{\downarrow} \models \delta(q_{\chi}, \sigma(k), \uparrow)$ if $D = [k]_{\sim}$, or such that $Q^k, Q^k_{\downarrow} \models \delta(q_{\chi}, \sigma(k), \nearrow)$ if $D \neq [k]_{\sim}$, we have

$$\{\langle q', D \rangle : q' \in Q^k\} \cup \{\langle q', [k]_{\sim} \rangle : q' \in Q^k_{\downarrow}\} \subseteq F_{k+1}^{\dagger}$$

By considering subruns starting with the sets of configurations in (iii) and (iv), and the inductive hypothesis, it follows that $\sigma, j \models_D \psi$, and $\sigma, k \models_D \chi$ for all $k \in \{i, \dots, j\}$, so $\sigma, i \models_D \psi R_{\chi}$ as required. If $\langle q_{\psi R_{\chi}}, D \rangle \in F_{j+1}^{\dagger}$ for all $j \geq i$, the argument is again simpler.

For case $\phi' = \downarrow \psi$, we have $\sigma, i \models_D \downarrow \psi$ iff $\sigma, i \models_{[i]_{\sim}} \psi$. By the inductive hypothesis, that is iff:

- (v) for some $Q^{\dagger}, Q^{\dagger}_{\downarrow} \subseteq Q$ such that $Q^{\dagger}, Q^{\dagger}_{\downarrow} \models \delta(q_{\psi}, \sigma(i), \uparrow)$, \mathcal{A}_{ϕ} has an infinite run from position $i+1$ of σ , starting with $\{\langle q', [i]_{\sim} \rangle : q' \in Q^{\dagger} \cup Q^{\dagger}_{\downarrow}\}$.

On the other hand, \mathcal{A}_{ϕ} having an infinite run from position $i+1$ of σ , starting with

$$\{\langle q', D \rangle : q' \in Q'\} \cup \{\langle q', [i]_{\sim} \rangle : q' \in Q'_{\downarrow}\}$$

for some $Q', Q'_{\downarrow} \subseteq Q$ such that $Q', Q'_{\downarrow} \models \delta(q_{\downarrow \psi}, \sigma(i), \uparrow)$ if $D = [i]_{\sim}$, or such that $Q', Q'_{\downarrow} \models \delta(q_{\downarrow \psi}, \sigma(i), \nearrow)$ if $D \neq [i]_{\sim}$, is equivalent to:

- (vi) for some $Q', Q'_{\downarrow} \subseteq Q$ such that $Q', Q'_{\downarrow} \models \delta(q_{\psi}, \sigma(i), \uparrow)[\downarrow q' / q' : q' \in Q]$, \mathcal{A}_{ϕ} has an infinite run from position $i+1$ of σ , starting with

$$\{\langle q', D \rangle : q' \in Q'\} \cup \{\langle q', [i]_{\sim} \rangle : q' \in Q'_{\downarrow}\}$$

It remains to observe that $Q', Q'_{\downarrow} \models \delta(q_{\psi}, \sigma(i), \uparrow)[\downarrow q' / q' : q' \in Q]$ iff $Q'_{\downarrow} = Q^{\dagger} \cup Q^{\dagger}_{\downarrow}$ for some $Q^{\dagger}, Q^{\dagger}_{\downarrow} \models \delta(q_{\psi}, \sigma(i), \uparrow)$, so (v) and (vi) are equivalent. \square

2.4 Counter Automata

We introduce below a class of nondeterministic automata on ω -words which have ε transitions and \mathbb{N} -valued counters. The set of counters of such an automaton will have structure: there will be a finite set called the basis of the automaton, and each counter will be a nonempty subset of the basis. In the course of a transition, the automaton will be able either to increment a counter, or to decrement a counter if nonzero, or to perform a simultaneous nondeterministic transfer with respect to a mapping f from counters to sets of counters. The latter transfers the value of each counter c to the counters in $f(c)$, nondeterministically splitting it. However, only mappings which satisfy a distributivity constraint in terms of the structure of the set of counters may be used.

The observation that simultaneous nondeterministic transfers arising from translating safety 1ARA₁ are distributive (cf. the proof of Theorem 3.2), and that distributivity enables nonemptiness of the counter automata to be decided in space exponential in basis size (cf. the proof of Theorem 3.3), are key components of the paper.

We shall only consider automata with no cycles of ε transitions, and they will recognise safety languages, so every infinite run will accept some ω -word.

The automata will be faulty in the sense that their counters may erroneously increase at any time.

Formally, for a finite set X and $C \subseteq \mathcal{P}(X) \setminus \{\emptyset\}$, let $L(C)$ be the set of all instructions:

- $\langle \text{inc}, c \rangle$ and $\langle \text{dec}, c \rangle$ for $c \in C$;
- $\langle \text{transf}, f \rangle$ for mappings $f : C \rightarrow \mathcal{P}(C)$ which are *distributive* as follows: whenever $c \in C$, $c \subseteq \bigcup_{i=1}^k c_i$, and $c'_i \in f(c_i)$ for each $i = 1, \dots, k$, there exists $c' \in f(c)$ such that $c' \subseteq \bigcup_{i=1}^k c'_i$.

A *safety powerset counter automaton with nondeterministic transfers and incrementing errors* (shortly, *safety IPCANT*) \mathcal{C} is a tuple $\langle \Sigma, Q, q_I, X, C, \delta \rangle$ such that:

- Σ is a finite alphabet;
- Q is a finite set of states, and q_I is the initial state;
- X is a finite set called the *basis*, and $C \subseteq \mathcal{P}(X) \setminus \{\emptyset\}$ is the set of counters;
- $\delta \subseteq Q \times (\Sigma \uplus \{\varepsilon\}) \times L(C) \times Q$ is a transition relation which does not contain a cycle of ε transitions.

A configuration of \mathcal{C} is a pair $\langle q, v \rangle$, where $q \in Q$ and v is a counter valuation, i.e. $v : C \rightarrow \mathbb{N}$. We say that $\langle q, v \rangle$ has an error-free transition labelled by $w \in \Sigma \uplus \{\varepsilon\}$ and performing $l \in L(C)$ to $\langle q', v' \rangle$, and we write $\langle q, v \rangle \xrightarrow{w, l}_{\checkmark} \langle q', v' \rangle$, iff $\langle q, w, l, q' \rangle \in \delta$ and v' can be obtained from v by l . The latter is defined as follows:

- instructions $\langle \text{inc}, c \rangle$ and $\langle \text{dec}, c \rangle$ have the standard interpretations, where $\langle \text{dec}, c \rangle$ is fireable iff $v(c) > 0$;
- v' can be obtained from v by $\langle \text{transf}, f \rangle$ iff there exist $K_{c'}^c \geq 0$ for each $c \in C$ and $c' \in f(c)$, such that:

$$\text{for each } c \in C, v(c) = \sum_{c' \in f(c)} K_{c'}^c \quad \text{for each } c' \in C, v'(c') = \sum_{f(c) \ni c} K_{c'}^c$$

in particular, $\langle \mathbf{transf}, f \rangle$ is fireable iff $v(c) = 0$ whenever $f(c) = \emptyset$.

For counter valuations v and v_\vee , we write $v \leq v_\vee$ iff, for all c , $v(c) \leq v_\vee(c)$. To allow transitions of \mathcal{C} to contain incrementing errors, we define $\langle q, v \rangle \xrightarrow{w, l} \langle q', v' \rangle$ to mean that there exist v_\vee and v'_\vee with $v \leq v_\vee$, $\langle q, v_\vee \rangle \xrightarrow{w, l}_\vee \langle q', v'_\vee \rangle$ and $v'_\vee \leq v'$.

We say that \mathcal{C} accepts an ω -word w over Σ iff \mathcal{C} has a run $\langle q_0, v_0 \rangle \xrightarrow{w_0, l_0} \langle q_1, v_1 \rangle \xrightarrow{w_1, l_1} \dots$ where $\langle q_0, v_0 \rangle$ is the initial configuration $\langle q_I, \mathbf{0} \rangle$ and $w = w_0 w_1 \dots$.

Example 2.6. Given $Y \subseteq X$, let $f_Y(c) = \emptyset$ if $c \cap Y \neq \emptyset$, and $f_Y(c) = \{c\}$ otherwise. Observe that f_Y is distributive. The instruction $\langle \mathbf{transf}, f_Y \rangle$ is fireable iff each counter which intersects Y is zero, and it does not change the value of any counter. Hence, we may write $\langle \mathbf{ifz}^\cap, Y \rangle$ instead of $\langle \mathbf{transf}, f_Y \rangle$.

Suppose $C = \{\{x\} : x \in X\}$, i.e. the set of counters has no structure. The instruction $\langle \mathbf{ifz}^\cap, Y \rangle$ is fireable iff each counter $\{x\}$ for $x \in Y$ is zero. Observe that every $f : C \rightarrow \mathcal{P}(C)$ is distributive. For instance, given $c \in C$ and nonempty $C' \subseteq C$, let $f_{c, C'}(c) = C'$ and $f_{c, C'}(c') = \{c'\}$ for $c' \neq c$. The instruction $\langle \mathbf{transf}, f_{c, C'} \rangle$ nondeterministically distributes the value of c to the counters in C' .

For \mathcal{C} as above, let us say that a transition $\langle q, v \rangle \xrightarrow{w, l} \langle q', v' \rangle$ is *lazy* iff either $\langle q, v \rangle \xrightarrow{w, l}_\vee \langle q', v' \rangle$, or l is of the form $\langle \mathbf{dec}, c \rangle$, $v(c) = 0$ and $v' = v$. Thus, in lazy transitions, only incrementing errors which enable decrements of counters with value 0 may occur. The following straightforward proposition shows that restricting to lazy transitions does not affect the languages of safety IPCANTs.

PROPOSITION 2.7. *Whenever $\langle q, v \rangle \xrightarrow{w, l} \langle q', v' \rangle$ is a transition of a safety IPCANT \mathcal{C} and $v_\dagger \leq v$, there exists a lazy transition $\langle q, v_\dagger \rangle \xrightarrow{w, l} \langle q', v'_\dagger \rangle$ of \mathcal{C} such that $v'_\dagger \leq v'$.*

A set L of ω -words over an alphabet Σ is called *safety* [Alpern and Schneider 1987] iff it is closed under limits of finite prefixes, i.e. for each ω -word w , if for each $i > 0$ there exists $w'_i \in L$ such that the i -prefixes of w and w'_i are equal, then $w \in L$. For each safety IPCANT, the tree of all its lazy runs is finitely branching, so by simplifying the argument in the proof of Proposition 2.2, and by Proposition 2.7, we obtain:

PROPOSITION 2.8. *The language of each safety IPCANT is safety.*

3. UPPER BOUND

This section contains a two-stage proof that nonemptiness of safety 1ARA_1 is in EXPSPACE . The first theorem below shows that each such automaton \mathcal{A} is translatable to a safety IPCANT $\mathcal{C}_\mathcal{A}$ of at most exponential size, but whose basis size is polynomially (in fact, linearly) bounded. Nonemptiness is preserved, since $\mathcal{C}_\mathcal{A}$ accepts exactly the string projections of data ω -words in the language of \mathcal{A} . By the second theorem, nonemptiness of $\mathcal{C}_\mathcal{A}$ is decidable in space exponential in its basis size and polynomial (in fact, polylogarithmic) in its alphabet size and number of states, so space exponential in the size of \mathcal{A} suffices overall.

We start with a piece of notation and a lemma about IPCANT. Suppose C is a set of counters over a basis X . For counter valuations v_\vee and v , let us write

$v_{\downarrow} \sqsubseteq v$ iff there exists $v_{\uparrow} \leq v$ which can be obtained from v_{\downarrow} by performing $\langle \mathbf{transf}, c \mapsto \{d : c \subseteq d\} \rangle$. The lemma states that \sqsubseteq is downwards compatible with every simultaneous nondeterministic transfer.

LEMMA 3.1. *Whenever $v_{\downarrow} \sqsubseteq v$ and v' is obtainable from v by some $\langle \mathbf{transf}, f \rangle$ with distributive f , there exists v'_{\downarrow} obtainable from v_{\downarrow} by $\langle \mathbf{transf}, f \rangle$ and such that $v'_{\downarrow} \sqsubseteq v'$.*

PROOF. We use the following shorthand: $\tilde{v} = \bigcup_{c \in C} \{\langle c, 1 \rangle, \dots, \langle c, v(c) \rangle\}$.

The assumptions are equivalent to existence of: an injective $\iota : \tilde{v}_{\downarrow} \rightarrow \tilde{v}$ such that $c \subseteq d$ whenever $\iota\langle c, i \rangle = \langle d, j \rangle$, and a bijective $\beta : \tilde{v} \rightarrow \tilde{v}'$ such that $f(d) \ni d'$ whenever $\beta\langle d, j \rangle = \langle d', j' \rangle$.

For each $\langle c, i \rangle \in \tilde{v}_{\downarrow}$, we have $c \subseteq d$ where $\iota\langle c, i \rangle = \langle d, j \rangle$, and $f(d) \ni d'$ where $\beta\langle d, j \rangle = \langle d', j' \rangle$, so by distributivity of f , there exists $c' \in f(c)$ such that $c' \subseteq d'$. Hence, there exist a counter valuation v'_{\downarrow} and a bijective $\beta_{\downarrow} : \tilde{v}_{\downarrow} \rightarrow \tilde{v}'_{\downarrow}$ such that $c' \in f(c)$ and $c' \subseteq d'$ whenever $\beta_{\downarrow}\langle c, i \rangle = \langle c', i' \rangle$ and $(\beta \circ \iota)\langle c, i \rangle = \langle d', j' \rangle$. It remains to observe that $\beta \circ \iota \circ \beta_{\downarrow}^{-1}$ is an injection from \tilde{v}'_{\downarrow} to \tilde{v}' . \square

THEOREM 3.2. *Given a safety 1ARA₁ \mathcal{A} , a safety IPCANT $\mathcal{C}_{\mathcal{A}}$ is computable in polynomial space, such that $\mathcal{C}_{\mathcal{A}}$ and \mathcal{A} have the same alphabet, the basis size of $\mathcal{C}_{\mathcal{A}}$ is linear in the number of states of \mathcal{A} , and $L(\mathcal{C}_{\mathcal{A}}) = \{\text{str}(\sigma) : \sigma \in L(\mathcal{A})\}$.*

PROOF. The proof is an adaptation of the proof of [Demri and Lazić 2009, Theorem 4.4], where it was shown how to translate in polynomial space weak 1ARA₁ to Büchi nondeterministic counter automata with ε transitions and incrementing errors, and whose instructions are increments, decrements and zero tests of individual counters. We show below essentially that, since \mathcal{A} is safety, zero tests of individual counters, cycles of ε transitions and the Büchi acceptance condition can be eliminated using nondeterministic transfers with a suitable basis and set of counters, resulting in a safety IPCANT.

Let $\mathcal{A} = \langle \Sigma, Q, q_I, \delta \rangle$. We first introduce an abstraction which maps a finite set F of configurations of \mathcal{A} at a position i of a data word σ over Σ to a triple $\langle a, Q_{\uparrow}, \# \rangle$ such that: $a = \sigma(i)$, Q_{\uparrow} is the set of all states that occur in F paired with $[i]_{\sim}$, and for each nonempty $R \subseteq Q$, $\#(R)$ is the number of data $D \neq [i]_{\sim}$ for which R is the set of all states that occur in F paired with D . Thus, the abstraction records only the letter at position i , and equalities among the datum at position i and data in configurations in F . We then observe that nonemptiness of \mathcal{A} is equivalent to existence of an infinite sequence of abstract transitions which starts from a triple of the form $\langle a, \{q_I\}, \mathbf{0} \rangle$. In other words, searching for a data ω -word σ over Σ and an infinite run of \mathcal{A} on σ can be performed one position at a time, while keeping in memory only the information recorded by the abstraction.

Formally, we define $H_{\mathcal{A}}$ to be the set of all triples $\langle a, Q_{\uparrow}, \# \rangle$ for which $a \in \Sigma$, $Q_{\uparrow} \subseteq Q$, and $\# : \mathcal{P}(Q) \setminus \{\emptyset\} \rightarrow \mathbb{N}$. For a data word σ over Σ , a position $0 \leq i < |\sigma|$, and finite set F of configurations, let $h(\sigma, i, F) = \langle \sigma(i), Q_{\uparrow}^{F, [i]_{\sim}}, \#^{F, [i]_{\sim}} \rangle$, where, for each nonempty $R \subseteq Q$:

$$Q_{\uparrow}^{F, D} = \{q : \langle q, D \rangle \in F\} \quad \#^{F, D}(R) = |\{D' \neq D : Q_{\uparrow}^{F, D'} = R\}|$$

To obtain a successor of a member of $H_{\mathcal{A}}$, for each configuration that it represents, sets of states which satisfy the appropriate positive Boolean formula in \mathcal{A} are chosen, and then two cases are distinguished: either the datum at the next position occurs in the next set of configurations, or not. Thus, we write $\langle a, Q_{\uparrow}, \sharp \rangle \rightarrow \langle a', Q'_{\uparrow}, \sharp' \rangle$ iff, for each $q \in Q_{\uparrow}$, there exist $Q^q, Q^q_{\downarrow} \models \delta(q, a, \uparrow)$, and for each nonempty $R \subseteq Q$, $j \in \{1, \dots, \sharp(R)\}$ and $q \in R$, there exist $Q^{R,j,q}, Q^{R,j,q}_{\downarrow} \models \delta(q, a, \uparrow)$, such that:

- either $\sharp' = \sharp^{\dagger}[Q'_{\uparrow} \mapsto \sharp^{\dagger}(Q'_{\uparrow}) - 1]$,
- or $Q'_{\uparrow} = \emptyset$ and $\sharp' = \sharp^{\dagger}$,

where, for each nonempty $R' \subseteq Q$, $\sharp^{\dagger}(R')$ is defined as

$$\begin{aligned} & |\{ \langle R, j \rangle : \bigcup_{q \in R} Q^{R,j,q} = R' \}| + \\ & \begin{cases} 1, & \text{if } \bigcup_{q \in Q_{\uparrow}} Q^q \cup \bigcup_{q \in Q_{\uparrow}} Q^q_{\downarrow} \cup \bigcup_{R,j} \bigcup_{q \in R} Q^{R,j,q}_{\downarrow} = R' \\ 0, & \text{otherwise} \end{cases} \end{aligned}$$

We claim the following correspondence between infinite sequences of transitions in $H_{\mathcal{A}}$ from initial triples and infinite runs of \mathcal{A} from initial configurations:

- (*) $\langle a_0, Q_{\uparrow}^0, \sharp_0 \rangle \rightarrow \langle a_1, Q_{\uparrow}^1, \sharp_1 \rangle \rightarrow \dots$ is an infinite sequence of transitions in $H_{\mathcal{A}}$ such that $Q_{\uparrow}^0 = \{q_I\}$ and $\sharp_0 = \mathbf{0}$ iff \mathcal{A} has an infinite run $F_0 \xrightarrow{\sigma, 0} F_1 \xrightarrow{\sigma, 1} \dots$ on a data ω -word σ over Σ such that $F_0 = \{\langle q_I, [0]_{\sim} \rangle\}$ and $\langle a_i, Q_{\uparrow}^i, \sharp_i \rangle = h(\sigma, i, F_i)$ for each $i \in \mathbb{N}$.

One direction is straightforward, since $h(\sigma, 0, \{\langle q_I, [0]_{\sim} \rangle\}) = \langle \sigma(0), \{q_I\}, \mathbf{0} \rangle$, and $F \xrightarrow{\sigma, i} F'$ implies $h(\sigma, i, F) \rightarrow h(\sigma, i+1, F')$. For the other direction, suppose $\langle a_0, Q_{\uparrow}^0, \sharp_0 \rangle \rightarrow \langle a_1, Q_{\uparrow}^1, \sharp_1 \rangle \rightarrow \dots$ is an infinite sequence of transitions in $H_{\mathcal{A}}$, $Q_{\uparrow}^0 = \{q_I\}$ and $\sharp_0 = \mathbf{0}$. For each $i \in \mathbb{N}$, let σ_i be a data word over Σ of length $i+1$ and F_i be a set of configurations for σ_i with $\langle a_i, Q_{\uparrow}^i, \sharp_i \rangle = h(\sigma_i, i, F_i)$, chosen as follows:

- We take $\text{str}(\sigma_0) = a_0$, $\sim^{\sigma_0} = \{\langle 0, 0 \rangle\}$, and $F_0 = \{\langle q_I, \{0\} \rangle\}$.
- Given σ_i and F_i , we choose σ_{i+1} and F_{i+1} for which σ_i is the $(i+1)$ -prefix of σ_{i+1} , $\langle a_{i+1}, Q_{\uparrow}^{i+1}, \sharp_{i+1} \rangle = h(\sigma_{i+1}, i+1, F_{i+1})$, and $F_i \xrightarrow{\sigma_i, i} F_{i+1}$.

Now, let σ^{\dagger} be the limit of the σ_i , i.e. such that for each $i \in \mathbb{N}$, σ_i is the $(i+1)$ -prefix of σ^{\dagger} . For each $i \in \mathbb{N}$, let F_i^{\dagger} be the unique set of configurations for σ^{\dagger} that satisfies

$$F_i = \{ \langle q, D \cap \{0, \dots, i\} \rangle : \langle q, D \rangle \in F_i^{\dagger} \}$$

Observe that $|F_i^{\dagger}| = |F_i|$, so F_i^{\dagger} is finite. Moreover, $h(\sigma^{\dagger}, i, F_i^{\dagger}) = h(\sigma_i, i, F_i)$, so $h(\sigma^{\dagger}, i, F_i^{\dagger}) = \langle a_i, Q_{\uparrow}^i, \sharp_i \rangle$. Finally, since $F_i \xrightarrow{\sigma_i, i} F_{i+1}$, we have $F_i^{\dagger} \xrightarrow{\sigma^{\dagger}, i} F_{i+1}^{\dagger}$.

The nondeterministic procedure below guesses an infinite sequence $\langle a_0, Q_{\uparrow}^0, \sharp_0 \rangle \rightarrow \langle a_1, Q_{\uparrow}^1, \sharp_1 \rangle \rightarrow \dots$ of transitions in $H_{\mathcal{A}}$ such that $Q_{\uparrow}^0 = \{q_I\}$ and $\sharp_0 = \mathbf{0}$ in the following manner: whenever the main loop has been performed i times and execution is at the end of step (2), a , Q_{\uparrow} and the counters c store a_i , Q_{\uparrow}^i and \sharp_i (respectively), and all the counters d have value 0. In the notation of the definition above of transitions in $H_{\mathcal{A}}$, each $d(R', R'_{\downarrow})$ is used to count the number of pairs $\langle R, j \rangle$ such that $\bigcup_{q \in R} Q^{R,j,q} = R'$ and $\bigcup_{q \in R} Q^{R,j,q}_{\downarrow} = R'_{\downarrow}$. If one or more choices in steps (3) or (4) are not possible, the procedure blocks.

- (0) Set $c(R) := 0$ for each nonempty $R \subseteq Q$, and $d(R, R_\downarrow) := 0$ for each $R, R_\downarrow \subseteq Q$.
- (1) Set $Q_\uparrow := \{q_I\}$.
- (2) Choose $a \in \Sigma$.
- (3) While $c(R) > 0$ for some nonempty $R \subseteq Q$, do:
 - decrement $c(R)$;
 - for each $q \in R$, choose $Q^q, Q_\downarrow^q \models \delta(q, a, \uparrow)$;
 - increment $d(\bigcup_{q \in R} Q^q, \bigcup_{q \in R} Q_\downarrow^q)$.
- (4) For each $q \in Q_\uparrow$, choose $Q^q, Q_\downarrow^q \models \delta(q, a, \uparrow)$.
- (5) Increment $c(\bigcup_{q \in Q_\uparrow} Q^q \cup \bigcup_{q \in Q_\uparrow} Q_\downarrow^q \cup \bigcup_{d(R, R_\downarrow) > 0} R_\downarrow)$.
- (6) While $d(R, R_\downarrow) > 0$ for some $R, R_\downarrow \subseteq Q$, decrement $d(R, R_\downarrow)$, and increment $c(R)$ if R is nonempty.
- (7) Either choose nonempty Q_\uparrow with $c(Q_\uparrow) > 0$ and decrement $c(Q_\uparrow)$, or $Q_\uparrow := \emptyset$.
- (8) Repeat from (2).

By (*), we have that the procedure has an infinite execution such that the letters chosen in step (2) are a_0, a_1, \dots iff \mathcal{A} accepts a data ω -word σ such that $a_i = \sigma(i)$ for each $i \in \mathbb{N}$. Therefore, in the remainder of the proof, we show that the procedure is implementable by a safety IPCANT $\mathcal{C}_\mathcal{A}$ which is computable in polynomial space and whose basis size is linear in $|Q|$.

For $R, R_\downarrow \subseteq Q$, let

$$\overline{R} = \{\overline{*}\} \cup \{\overline{q} : q \in R\} \quad \overline{\overline{R, R_\downarrow}} = \{\overline{*}\} \cup \{\overline{\overline{q}} : q \in R\} \cup \{\overline{\overline{q_\downarrow}} : q \in R_\downarrow\}$$

We define the basis of $\mathcal{C}_\mathcal{A}$ as $\overline{Q} \cup \overline{\overline{Q, Q}}$ (where we assume disjointness), and the counters of $\mathcal{C}_\mathcal{A}$ are: \overline{R} for each $R \subseteq Q$, and $\overline{\overline{R, R_\downarrow}}$ for each $R, R_\downarrow \subseteq Q$. The set of counters of $\mathcal{C}_\mathcal{A}$ is thus essentially $\mathcal{P}(Q) \cup \mathcal{P}(Q)^2$. Note that, compared to the procedure above, $\mathcal{C}_\mathcal{A}$ has the extra counter $\overline{\emptyset}$.

The states of $\mathcal{C}_\mathcal{A}$ are used for control, and for storing the letters from Σ as well as the elements and subsets of Q . Step (0) is implemented by default, and steps (1), (2), (4) and (8) are straightforward.

Step (3) can be performed by a single simultaneous nondeterministic transfer, with the mapping

$$\begin{aligned} \overline{R} &\mapsto \{\overline{\bigcup_{q \in R} Q^q, \bigcup_{q \in R} Q_\downarrow^q} : \forall q \in R (Q^q, Q_\downarrow^q \models \delta(q, a, \uparrow))\}, \\ \overline{\overline{R, R_\downarrow}} &\mapsto \{\overline{\overline{R, R_\downarrow}} : R, R_\downarrow \subseteq Q\} \end{aligned}$$

whose distributivity is a key component of the paper. To show that it holds, suppose $\overline{R} \subseteq \bigcup_{i=1}^k \overline{R^i}$, and $Q^{i,q}, Q_\downarrow^{i,q} \models \delta(q, a, \uparrow)$ for each $i \in \{1, \dots, k\}$ and $q \in R^i$. Given $q \in R$, let i_q be such that $q \in R^{i_q}$. We then have, as required:

$$\overline{\bigcup_{q \in R} Q^{i_q, q}, \bigcup_{q \in R} Q_\downarrow^{i_q, q}} \subseteq \bigcup_{i=1}^k \overline{\overline{Q^{i,q}, Q_\downarrow^{i,q}}}$$

The following is an implementation of step (5):

—Set $R' := \bigcup_{q \in Q_\uparrow} Q^q \cup \bigcup_{q \in Q_\uparrow} Q_\downarrow^q$.

- For each $q \in Q$, either perform the transfer that verifies that each $\overline{\overline{R}}, \overline{R}_\downarrow$ with $q \in R_\downarrow$ is zero (cf. Example 2.6), or choose $R, R_\downarrow \subseteq Q$ with $q \in R_\downarrow$, decrement $\overline{\overline{R}}, \overline{R}_\downarrow$, increment $\overline{\overline{R}}, \overline{R}_\downarrow$ and set $R' := R' \cup \{q\}$.
- Increment $\overline{R'}$.

For step (6), we use the transfer with the mapping

$$\{\overline{R} \mapsto \{\overline{R}\}, \overline{\overline{R}}, \overline{R}_\downarrow \mapsto \{\overline{R}\} : R, R_\downarrow \subseteq Q\}$$

which is distributive since $\overline{\overline{R}}, \overline{R}_\downarrow \subseteq \bigcup_{i=1}^k \overline{R^i}, \overline{R}_\downarrow^i$ implies $\overline{R} \subseteq \bigcup_{i=1}^k \overline{R^i}$.

Finally, in step (7), if $Q_\uparrow := \emptyset$ is performed, then either $\overline{\emptyset}$ is decremented or not.

Observe therefore that the auxiliary counter $\overline{\emptyset}$ is transferred to $\overline{\overline{\emptyset}}, \overline{\emptyset}$ in step (3), that $\overline{\overline{\emptyset}}, \overline{\emptyset}$ is transferred to $\overline{\emptyset}$ in step (6), and that those two counters do not affect anything else.

In step (2), $\mathcal{C}_\mathcal{A}$ performs an a transition, and all other transitions are ε . However, the only cycle in the transition graph of $\mathcal{C}_\mathcal{A}$ corresponds to the loop (2)–(8), so the requirement of no cycles of ε transitions is met.

The only nontrivial aspect of computing $\mathcal{C}_\mathcal{A}$ in space polynomial in the size of \mathcal{A} is the implementation of step (3). However, for each $R \subseteq Q$, the set

$$\{\overline{\bigcup_{q \in R} Q^q}, \overline{\bigcup_{q \in R} Q_\downarrow^q} : \forall q \in R(Q^q, Q_\downarrow^q \models \delta(q, a, \uparrow))\}$$

can be output by iterating over all mappings $q \mapsto \langle Q^q, Q_\downarrow^q \rangle$ from R to $\mathcal{P}(Q)^2$. Each such mapping can be stored in space $2|Q|^2$, and deciding $Q^q, Q_\downarrow^q \models \delta(q, a, \uparrow)$ amounts to evaluating a propositional formula.

It remains to show that incrementing errors cannot cause $\mathcal{C}_\mathcal{A}$ to accept an ω -word $a_0 a_1 \dots$ which it does not accept without incrementing errors. Informally, that is the case because incrementing errors in runs of $\mathcal{C}_\mathcal{A}$ amount to introductions of spurious threads into corresponding runs of \mathcal{A} , which can only make acceptance harder.

Suppose $\mathcal{C}_\mathcal{A}$ accepts an ω -word $a_0 a_1 \dots$, i.e. the implementation of the procedure above has an infinite execution E which may contain incrementing errors and which chooses in step (2) the letters a_0, a_1, \dots . Below, we define an error-free infinite execution E_\surd such that the letters chosen in step (2) are also a_0, a_1, \dots , and we show by induction that the following are satisfied before each step:

- (i) $v_\surd \sqsubseteq v$ (cf. Lemma 3.1), where v and v_\surd are the current counter valuations in E and E_\surd (respectively);
- (ii) $Q_\uparrow^\surd \subseteq Q_\uparrow$, if Q_\uparrow and Q_\uparrow^\surd are defined, where they are the current values of the variable in E and E_\surd (respectively).

Initially, we have that v and v_\surd equal $\mathbf{0}$, and that Q_\uparrow and Q_\uparrow^\surd are undefined, so the inductive base is trivial. We also have that $v_\surd \sqsubseteq v$ and $v \leq v'$ imply $v_\surd \sqsubseteq v'$, i.e. the \sqsubseteq relation is preserved by incrementing errors in the second argument.

Steps (1) and (2). E_\surd performs the same transitions as E .

Steps (3) and (6). E_\surd performs the transfers as in Lemma 3.1.

Step (4). For each $q \in Q_{\uparrow}^{\vee} \subseteq Q_{\uparrow}$, the same Q^q and Q_{\downarrow}^q are chosen in E_{\vee} as in E .

Step (5). For each $q \in Q$, if there exist R^{\vee} and $R_{\downarrow}^{\vee} \ni q$ such that $v_{\vee}(\overline{R^{\vee}}, \overline{R_{\downarrow}^{\vee}}) > 0$, we have by (i) that there exist R and $R_{\downarrow} \ni q$ such that $v(\overline{R}, \overline{R_{\downarrow}}) > 0$. It follows that $R'_{\vee} \subseteq R'$, where R' is the value of the variable after the implementation of step (5) is executed in E , and R'_{\vee} is the value after the unique error-free execution in E_{\vee} . Hence, (i) is preserved.

Step (7). Let $\iota : \widetilde{v_{\vee}} \rightarrow \widetilde{v}$ be an injection (cf. the proof of Lemma 3.1), and Q_{\uparrow} be the value chosen in E . If $\overline{Q_{\uparrow}}$ is decremented and $\iota(\overline{Q_{\uparrow}^{\vee}}, i) = \langle \overline{Q_{\uparrow}}, j \rangle$ for some $\overline{Q_{\uparrow}^{\vee}}$, i and j (in particular, $Q_{\uparrow}^{\vee} \subseteq Q_{\uparrow}$), then choose such Q_{\uparrow}^{\vee} in E_{\vee} and decrement $\overline{Q_{\uparrow}^{\vee}}$. Otherwise, choose \emptyset in E_{\vee} without decrementing.

That completes the definition of E_{\vee} and the proof. \square

THEOREM 3.3. *Nonemptiness of safety IPCANT is decidable in space exponential in basis size and polylogarithmic in alphabet size and number of locations.*

PROOF. Suppose $\mathcal{C} = \langle \Sigma, Q, q_I, X, C, \delta \rangle$ is a safety IPCANT. By Proposition 2.7, \mathcal{C} is nonempty iff it has an infinite sequence of lazy transitions from the initial configuration.

We define positive integers α_i and U_i for $i = 0, \dots, |X|$ as follows:

$$\alpha_0 = |Q| \quad U_0 = 1 \quad \alpha_{i+1} = 2(|X| - i)\alpha_i U_i^{|C|} \quad U_{i+1} = 3\alpha_i U_i^{|C|}$$

Let $m = 2\alpha_{|X|} U_{|X|}^{|C|}$. We shall show:

- (I) If \mathcal{C} has a sequence of $m - 1$ lazy transitions from the initial configuration, then it has an infinite sequence.

Therefore, nonemptiness of \mathcal{C} can be decided nondeterministically by guessing a sequence of $m - 1$ lazy transitions from the initial configuration. In every such sequence, each transition increases the sum of all counters by at most 1, so no counter can exceed $m - 1$. Since $m < 2^{2^{|X|^2 + |X|} \log(3|Q|)}$ and $|C| < 2^{|X|}$, a single configuration can be stored in space $2^{O(|X|^2)} O(\log |Q|)$. To guess a sequence of length $m - 1$, it suffices to store at most two configurations, the number of transitions guessed so far, and a fixed number of variables bounded by $|C| = 2^{O(|X|)} O(|\Sigma| \cdot |Q|)$ for indexing the transition relation of \mathcal{C} . Hence, nonemptiness of \mathcal{C} is decidable nondeterministically in space $2^{O(|X|^2)} O(\log(|\Sigma| \cdot |Q|))$, so by Savitch's Theorem, there is a deterministic algorithm of space complexity $2^{O(|X|^2)} O(\log(|\Sigma| \cdot |Q|)^2)$.

To show (I), suppose \mathcal{C} has a sequence of lazy transitions $S = \langle q_1, v_1 \rangle \xrightarrow{w_1, l_1} \dots \xrightarrow{w_{m-1}, l_{m-1}} \langle q_m, v_m \rangle$ from the initial configuration, but no infinite sequence. By careful repeated uses of the pigeonhole principle and the distributivity of simultaneous nondeterministic transfers, we shall obtain the contradiction that S must contain two equal configurations. To start with, some state must occur among q_1, \dots, q_m at least $m/|Q|$ times, so let $q \in Q$ and $J_0 \subseteq \{1, \dots, m\}$ be such that $|J_0| = m/\alpha_0 U_0^{|C|}$ and $q_j = q$ for each $j \in J_0$. We claim:

- (II) There exist an enumeration $x_1, \dots, x_{|X|}$ of X , and for $i = 1, \dots, |X|$, mappings $u_i : C_i \rightarrow \{0, \dots, U_i - 1\}$ where $C_i = \{c \in C : x_i \in c \wedge x_1, \dots, x_{i-1} \notin c\}$, and subsets J_i of $\{1, \dots, m\}$ of size $m/\alpha_i U_i^{|C|}$, such that the following property holds for each $0 \leq i \leq |X|$: for all $j \in J_i$, we have that $q_j = q$ and that for all $1 \leq i' \leq i$ and $c \in C_{i'}$, $v_j(c) = u_{i'}(c)$.

We establish (II) by proving the property inductively on i and simultaneously picking x_i , u_i and J_i . The case $i = 0$ is trivial. Assume that $0 \leq i < |X|$ and that $x_{i'}$, $u_{i'}$ and $J_{i'}$ for $i' = 1, \dots, i$ have been picked so that the property holds for i . Let us call a subsequence of S an i -subsequence iff there exist consecutive $j, j' \in J_i$ (i.e. where there is no $j'' \in J_i$ with $j < j'' < j'$) such that the subsequence begins at $\langle q_j, v_j \rangle$ and ends at $\langle q_{j'}, v_{j'} \rangle$. Let $J_i' \subseteq J_i$ consist of the beginning positions of the $|J_i|/2 = m/2\alpha_i U_i^{|C|}$ shortest i -subsequences. The length of the longest of those i -subsequences must be at most $2\alpha_i U_i^{|C|}$, since otherwise there would be at least $|J_i|/2$ i -subsequences of length more than $m/(|J_i|/2)$. Let $S^\dagger = \langle q_j, v_j \rangle \xrightarrow{w_j, l_j} \dots \xrightarrow{w_{j'-1}, l_{j'-1}} \langle q_{j'}, v_{j'} \rangle$ be an i -subsequence with $j \in J_i'$. We have $j' - j \leq 2\alpha_i U_i^{|C|}$, $q_j = q_{j'} = q$, and for all $1 \leq i' \leq i$ and $c \in C_{i'}$, $v_j(c) = v_{j'}(c) = u_{i'}(c)$. Recalling that $u_{i'} : C_{i'} \rightarrow \{0, \dots, U_{i'} - 1\}$, we obtain $\sum_{i'=1}^i \sum_{c \in C_{i'}} v_{j'}(c) \leq \sum_{i'=1}^i |C_{i'}| U_{i'}$.

To make progress, we prove:

- (III) There exists $x'_j \neq x_1, \dots, x_i$ such that, for each c with $x'_j \in c$ and $x_1, \dots, x_i \notin c$, $v_j(c) \leq 2\alpha_i U_i^{|C|} + \sum_{i'=1}^i |C_{i'}| U_{i'}$.

Suppose the contrary: for each $x' \neq x_1, \dots, x_i$, there exists $c_{x'}$ such that $x' \in c_{x'}$, $x_1, \dots, x_i \notin c_{x'}$, and $v_j(c_{x'}) > 2\alpha_i U_i^{|C|} + \sum_{i'=1}^i |C_{i'}| U_{i'}$. Let H be a directed acyclic graph on $\{j, \dots, j'\} \times C$, defined by letting the successors of $\langle k, d \rangle$ be:

- \emptyset , if $k = j'$;
- $\{\langle k+1, d' \rangle : d' \in f(d)\}$, if l_k is of the form $\langle \text{transf}, f \rangle$;
- $\{\langle k+1, d \rangle\}$, otherwise.

Now, for $c \in C$ and $k \in \{j, \dots, j'\}$, let $H(c, k)$ be the set of all d such that $\langle k, d \rangle$ is reachable in H from $\langle j, c \rangle$. We have $\sum_{d \in H(c, k)} v_k(d) \geq v_j(c) - (k - j)$ by induction on k . In particular, for each $x' \neq x_1, \dots, x_i$, we have $\sum_{d \in H(c_{x'}, j')} v_{j'}(d) \geq v_j(c_{x'}) - (j' - j) > \sum_{i'=1}^i |C_{i'}| U_{i'} \geq \sum_{i'=1}^i \sum_{c \in C_{i'}} v_{j'}(c)$, so there is some $d_{x'} \in H(c_{x'}, j')$ such that $x_1, \dots, x_i \notin d_{x'}$. Let $H_{x'}$ be a path in H from $\langle j, c_{x'} \rangle$ to $\langle j', d_{x'} \rangle$. For $k \in \{j, \dots, j'\}$, let $H_{x'}(k)$ denote the counter at position k in $H_{x'}$.

Consider any c with $x_1, \dots, x_i \notin c$. Observe that $c \subseteq \bigcup \{c_{x'} : x' \in c\}$. Let H_c be a path in H from $\langle j, c \rangle$, obtained as follows. Assuming that $k \in \{j, \dots, j' - 1\}$ and $H_c(k) \subseteq \bigcup \{H_{x'}(k) : x' \in c\}$:

- if l_k is of the form $\langle \text{transf}, f \rangle$, by distributivity of f and the definition of H , we can pick $H_c(k+1) \subseteq \bigcup \{H_{x'}(k+1) : x' \in c\}$;
- otherwise, we have $H_{x'}(k+1) = H_{x'}(k)$ for each $x' \in c$, and the only possibility is $H_c(k+1) = H_c(k)$.

Since $H_c(j') \subseteq \bigcup \{H_{x'}(j') : x' \in c\}$, we conclude that $x_1, \dots, x_i \notin H_c(j')$.

Using the paths H_c , we now show that, from the final configuration of S^\dagger , the instructions in S^\dagger can be performed repeatedly to obtain an infinite sequence of lazy transitions, which is a contradiction, so (III) holds. More precisely, since $v_j(d) = v_{j'}(d)$ for all $1 \leq i' \leq i$ and $d \in C_{i'}$, and $H_c(j) = c$ for all c , by (IV) below from $v_{j'}$ for $k = j, \dots, j' - 1$, there exist lazy transitions $\langle q_j, v_{j'} \rangle \xrightarrow{w_j, l_j} \dots \xrightarrow{w_{j'-1}, l_{j'-1}} \langle q_{j'}, v_{j'} \rangle$ such that $v_{j'}(d) \leq v_{j'}(d)$ for all $d \notin \{H_c(j') : x_1, \dots, x_i \notin c\}$. But $\{H_c(j') : x_1, \dots, x_i \notin c\} \subseteq \{c : x_1, \dots, x_i \notin c\}$, so (IV) can be applied from $v_{j'}$ for $k = j, \dots, j' - 1$, etc.

(IV) Suppose $k \in \{j, \dots, j' - 1\}$, and v'_k is a counter valuation such that $v'_k(d) \leq v_k(d)$ for all $d \notin \{H_c(k) : x_1, \dots, x_i \notin c\}$. There exists a lazy transition $\langle q_k, v'_k \rangle \xrightarrow{w_k, l_k} \langle q_{k+1}, v'_{k+1} \rangle$ such that $v'_{k+1}(d) \leq v_{k+1}(d)$ for all $d \notin \{H_c(k+1) : x_1, \dots, x_i \notin c\}$.

To show (IV), we distinguish between two cases:

—If l_k is of the form $\langle \text{transf}, f \rangle$, let $K_{d'}^d \geq 0$ for each $d \in C$ and $d' \in f(d)$ satisfy

$$\begin{aligned} &\text{for each } d \in C, v_k(d) = \sum_{d' \in f(d)} K_{d'}^d \\ &\text{for each } d' \in C, v_{k+1}(d') = \sum_{f(d) \ni d'} K_{d'}^d \end{aligned}$$

For $d \in C$ such that $v'_k(d) \leq v_k(d)$, pick any $K_{d'}^d \geq 0$ such that $v'_k(d) = \sum_{d' \in f(d)} K_{d'}^d$ and $K_{d'}^d \leq K_{d'}^d$ for each $d' \in f(d)$. For $d \in C$ such that $v'_k(d) > v_k(d)$, we have $d = H_c(k)$ for some c with $x_1, \dots, x_i \notin c$, so we can set $K_{d'}^d = K_{d'}^d$ for all $d' \in f(d) \setminus \{H_c(k+1)\}$, and $K_{H_c(k+1)}^d = K_{H_c(k+1)}^d + v'_k(d) - v_k(d)$. Now, for each $d' \in C$, let $v'_{k+1}(d') = \sum_{f(d) \ni d'} K_{d'}^d$, so that $\langle q_k, v'_k \rangle \xrightarrow{w_k, l_k} \langle q_{k+1}, v'_{k+1} \rangle$ lazily. Since $K_{d'}^d > K_{d'}^d$ implies $d' \in \{H_c(k+1) : x_1, \dots, x_i \notin c\}$, we have $v'_{k+1}(d') \leq v_{k+1}(d')$ for all $d' \notin \{H_c(k+1) : x_1, \dots, x_i \notin c\}$.

—Otherwise, v'_{k+1} is uniquely determined by the lazy transition $\langle q_k, v'_k \rangle \xrightarrow{w_k, l_k} \langle q_{k+1}, v'_{k+1} \rangle$, and has the required property as $H_c(k+1) = H_c(k)$ for all c .

For each $j \in J'_i$, let $x'_j \neq x_1, \dots, x_i$ be as in (III). For each c with $x'_j \in c$ and $x_1, \dots, x_i \notin c$, we have $v_j(c) < U_{i+1}$. Let x_{i+1} be such that there exists $J''_i \subseteq J'_i$ of size $|J''_i|/(|X| - i) = m/\alpha_{i+1}$ with $x_{i+1} = x'_j$ for all $j \in J''_i$. Thus, for all $j \in J''_i$ and $c \in C_{i+1}$, we have $v_j(c) < U_{i+1}$. Then let $u_{i+1} : C_{i+1} \rightarrow \{0, \dots, U_{i+1} - 1\}$ be such that there exists $J_{i+1} \subseteq J''_i$ of size $m/\alpha_{i+1} U_{i+1}^{|C|}$ with $v_j(c) = u_{i+1}(c)$ for all $j \in J_{i+1}$ and $c \in C_{i+1}$. That completes the inductive proof of (II).

Since $m = 2\alpha_{|X|} U_{|X|}^{|C|}$, we have from (II) that S contains two equal configurations, so \mathcal{C} has an infinite sequence of lazy transitions from the initial configuration. That is a contradiction, so (I) is shown. \square

By Theorems 3.2, 3.3 and 2.5, we obtain:

COROLLARY 3.4. *Safety 1ARA₁ nonemptiness and safety LTL₁[↓](X, R) satisfiability are in EXPSpace.*

4. LOWER BOUND

THEOREM 4.1. *Safety $1ARA_1$ nonemptiness and safety $LTL_1^\downarrow(\mathbf{X}, \mathbf{R})$ satisfiability are EXPSPACE-hard.*

PROOF. By Theorem 2.5, it suffices to show EXPSPACE-hardness of satisfiability for safety $LTL_1^\downarrow(\mathbf{X}, \mathbf{R})$. We shall reduce from the halting problem for Turing machines with exponentially long tapes. More precisely, a Turing machine \mathcal{M} is a tuple $\langle \Sigma, a_B, Q, q_I, \delta \rangle$ such that:

- Σ is a finite alphabet, and $a_B \in \Sigma$ denotes the blank symbol;
- Q is a finite set of states, and $q_I \in Q$ is the initial state;
- $\delta : Q \times \Sigma \rightarrow Q \times \Sigma \times \{-1, 1\}$ is the transition function.

If the size of \mathcal{M} is n , we consider its computation on a tape of length 2^n . More formally, a configuration of \mathcal{M} is of the form $\langle q, i, w \rangle$ where $q \in Q$ is the machine state, $0 \leq i < 2^n$ is the head position, and $w \in \Sigma^{2^n}$ is the tape contents. The initial configuration is $\langle q_I, 0, a_B^{2^n} \rangle$. A configuration $\langle q, i, w \rangle$ has a transition iff $0 \leq i + o < 2^n$ where $\langle q', a, o \rangle = \delta(q, w(i))$. In that case, we write $\langle q, i, w \rangle \rightarrow \langle q', i + o, w[i \mapsto a] \rangle$. Since \mathcal{M} can halt by requesting to move the head off an edge of the tape, it does not need to have a special halting state.

The following problem is EXPSPACE-complete: given $\mathcal{M} = \langle \Sigma, a_B, Q, q_I, \delta \rangle$ of size n , is the computation from the initial configuration with tape length 2^n infinite? (To reduce in polynomial time from the same problem with tape length 2^{n^k} , extend the machine by unreachable states until it is of size n^k .) We shall show that a sentence $\phi_{\mathcal{M}}$ of safety $LTL_1^\downarrow(\mathbf{X}, \mathbf{R})$ is computable in space logarithmic in n , such that the answer to the decision problem is ‘yes’ iff $\phi_{\mathcal{M}}$ is satisfiable.

Let $\widehat{\Sigma} = \{\widehat{a} : a \in \Sigma\}$. The alphabet of $\phi_{\mathcal{M}}$ is $\widetilde{\Sigma} = Q \uplus \{0_d, 1_d : d \in \{1, \dots, n\}\} \uplus \Sigma \uplus \widehat{\Sigma}$. To encode a tape cell, we write its position in binary followed by its contents. A configuration $\langle q, i, w \rangle$ is then encoded by the word below, where $\widehat{\Sigma}$ is used to mark the contents at head position. Let $w(i, i) = \widehat{w(i)}$, and $w(j, i) = w(j)$ for $j \neq i$.

$$q 0_1 \cdots 0_{n-1} 0_n w(0, i) 0_1 \cdots 0_{n-1} 1_n w(1, i) \cdots 1_1 \cdots 1_{n-1} 1_n w(2^n - 1, i)$$

The computation of \mathcal{M} from the initial configuration with tape length 2^n is infinite iff there exists a data ω -word σ over $\widetilde{\Sigma}$ such that:

- (i) $\text{str}(\sigma)$ is a sequence of encodings of configurations of \mathcal{M} ;
- (ii) $\text{str}(\sigma)$ begins with the encoding of the initial configuration $\langle q_I, 0, a_B^{2^n} \rangle$;
- (iii) for every two consecutive encodings in $\text{str}(\sigma)$ of configurations $\langle q, i, w \rangle$ and $\langle q', i', w' \rangle$, we have $\langle q, i, w \rangle \rightarrow \langle q', i', w' \rangle$.

Hence, it suffices to construct $\phi_{\mathcal{M}}$ such that σ satisfies $\phi_{\mathcal{M}}$ iff (i)–(iii) hold and:

- (iv) for every encoding in σ of a tape cell, all the letters b_d and $w(j, i)$ are in the same class;
- (v) for every two encodings in σ of tape cells with positions j and j' (occurring in one or two configuration encodings), their classes are the same iff $j = j'$.

The purpose of (iv) and (v) is to enable navigation through σ for checking (i)–(iii) in $\phi_{\mathcal{M}}$, whose size will be only polynomial in n .

For (i), we can split it into the following constraints, each of which is straightforward to express:

- the first letter is a state of \mathcal{M} ;
- every state of \mathcal{M} is succeeded by $0_1 \cdots 0_{n-1} 0_n$;
- every b_n is succeeded by an element of $\Sigma \uplus \widehat{\Sigma}$;
- for every b_d not succeeded by $1_{d+1} \cdots 1_n$, b_d occurs $n + 1$ positions later (the next position has the same binary digit d);
- for every 0_d succeeded by $1_{d+1} \cdots 1_n$, $1_d 0_{d+1} \cdots 0_n$ occurs $n + 1$ positions later (the next position has the opposite binary digit d);
- $1_1 \cdots 1_{n-1} 1_n$ followed by an element of $\Sigma \uplus \widehat{\Sigma}$ are succeeded by a state of \mathcal{M} ;
- between every two consecutive occurrences of states of \mathcal{M} , there is exactly one occurrence of an element of $\widehat{\Sigma}$.

Properties (ii) and (iv) are also straightforward. Before (iii), let us consider (v), which is equivalent to the following conjunction:

- (v.1) for every two encodings of tape cells, if their classes are the same then their positions are the same;
- (v.2) for every encoding of a tape cell, some tape cell in the next configuration encoding has the same class.

The more involved is (v.1). It amounts to requiring that, for all $d \in \{1, \dots, n\}$ and $b \in \{0, 1\}$, it is not the case that there is an occurrence of b_d and a subsequent occurrence of $(1 - b)_d$ with the same datum:

$$\bigwedge_{d=1}^n \bigwedge_{b=0}^1 \mathbf{G}(\overline{b_d} \vee \downarrow \mathbf{XG}(\overline{(1-b)_d} \vee \uparrow))$$

where \overline{a} abbreviates $\bigvee \{a' : a' \in \widehat{\Sigma} \setminus \{a\}\}$.

Property (iii) is now equivalent to asserting that the following hold for all $q \in Q$ and $a \in \Sigma$, where $\langle q', a', o \rangle = \delta(q, a)$:

- (iii.1) whenever q occurs with \widehat{a} in the same configuration encoding, the next occurrence of a state of \mathcal{M} is q' ;
- (iii.2) for every occurrence of some $b \in \Sigma$ in a configuration encoding which contains q and \widehat{a} , the next occurrence in the same class of an element of $\Sigma \uplus \widehat{\Sigma}$ is an occurrence of b or \widehat{b} ;
- (iii.3) for every occurrence of \widehat{a} in a configuration encoding containing q , the next occurrence in the same class of an element of $\Sigma \uplus \widehat{\Sigma}$ is an occurrence of a' , and n positions earlier (if $o = -1$) or later (if $o = 1$) an element of $\widehat{\Sigma}$ occurs.

The most involved is (iii.3), and the two cases of $o = -1$ and $o = 1$ are similar. Letting $\widehat{\Sigma}$ and $\widetilde{\Sigma}$ abbreviate $\bigvee \{b : b \in \widehat{\Sigma}\}$ and $\bigvee \{b : b \in \widetilde{\Sigma} \setminus \widehat{\Sigma}\}$ (respectively), (iii.3) with $o = -1$ is expressed by:

$$\mathbf{G}\left(q \Rightarrow \neg\left(\widetilde{\Sigma}\mathbf{U}\left(\widehat{a} \wedge \downarrow \mathbf{X}\left(\widetilde{\Sigma}\mathbf{U}\left(\widehat{\Sigma} \wedge \mathbf{X}^n \neg(a' \wedge \uparrow)\right)\right)\right)\right)\right)$$

To obtain a sentence of safety $\text{LTL}_1^\downarrow(\mathbf{X}, \mathbf{R})$ in the strict sense, we convert to negation normal form:

$$\mathbf{G}\left(\overline{q} \vee \left(\widehat{\Sigma}\mathbf{R}\left(\widehat{a} \vee \downarrow\mathbf{X}\left(\widehat{\Sigma}\mathbf{R}\left(\overline{\Sigma} \vee \mathbf{X}^n(a' \wedge \uparrow)\right)\right)\right)\right)\right)$$

To output $\widetilde{\Sigma}$ and $\phi_{\mathcal{M}}$ given \mathcal{M} as above, a fixed number of counters which are bounded by n suffice. \square

5. INCLUSION AND REFINEMENT

Using well-quasi-orderings, the proofs of Theorems 3.2 and 3.3, and that satisfiability over finite data words for $\text{LTL}_1^\downarrow(\mathbf{X}, \mathbf{F})$ is not primitive recursive [Demri and Lazić 2009, Theorem 5.2], we obtain the result below.

We remark that, in a similar manner, one can show that the following “model-checking” problems are decidable and not primitive recursive: whether the language of a Büchi one-way nondeterministic register automaton (with any number of registers) is included in the language of a safety 1ARA_1 or a safety $\text{LTL}_1^\downarrow(\mathbf{X}, \mathbf{R})$ sentence.

THEOREM 5.1. *The following problems are decidable and not primitive recursive:*

- inclusion for safety 1ARA_1 ;
- refinement for safety $\text{LTL}_1^\downarrow(\mathbf{X}, \mathbf{R})$.

PROOF. By Theorem 2.5, it suffices to establish that inclusion for safety 1ARA_1 is decidable and that refinement for safety $\text{LTL}_1^\downarrow(\mathbf{X}, \mathbf{R})$ is not primitive recursive.

For the former, suppose $\mathcal{A}_1 = \langle \Sigma, Q_1, q_I^1, \delta_1 \rangle$ and $\mathcal{A}_2 = \langle \Sigma, Q_2, q_I^2, \delta_2 \rangle$ are safety 1ARA_1 , where we need to determine whether $L(\mathcal{A}_1) \subseteq L(\mathcal{A}_2)$.

Let $\overline{\mathcal{A}}_2 = \langle \Sigma, Q_2, q_I^2, \overline{\delta}_2 \rangle$ be the dual automaton to \mathcal{A}_2 , so that each formula $\overline{\delta}_2(r, a, ?)$ is the dual to $\delta_2(r, a, ?)$, i.e. obtained by replacing every \top with \perp , every \wedge with \vee , and vice versa. Let $L(\overline{\mathcal{A}}_2)$ denote the language of $\overline{\mathcal{A}}_2$ with respect to *co-safety* acceptance: a data ω -word σ over Σ is in $L(\overline{\mathcal{A}}_2)$ iff \mathcal{A}_2 has a finite run $F_0 \xrightarrow{\sigma, 0} F_1 \xrightarrow{\sigma, 1} \dots \emptyset$ where $F_0 = \{\langle q_I^2, [0] \sim \rangle\}$. Considering \mathcal{A}_2 (resp., $\overline{\mathcal{A}}_2$) as a weak alternating automaton whose every state is of even (resp., odd) parity, we have by [Löding and Thomas 2000, Theorem 1] that $L(\overline{\mathcal{A}}_2)$ is the complement of $L(\mathcal{A}_2)$.

Now, let \mathcal{A}_\cap be the automaton for the intersection of \mathcal{A}_1 and $\overline{\mathcal{A}}_2$, obtained by adding a new initial state. More precisely, assuming that Q_1 and Q_2 are disjoint and do not contain q_I , let $\mathcal{A}_\cap = \langle \Sigma, \{q_I\} \cup Q_1 \cup Q_2, q_I, \delta_\cap \rangle$, where

$$\delta_\cap = \{\langle q_I, a, ? \rangle \mapsto \delta_1(q_I^1, a, ?) \wedge \overline{\delta}_2(q_I^2, a, ?) : a \in \Sigma, ? \in \{\uparrow, \nearrow\}\} \cup \delta_1 \cup \overline{\delta}_2$$

The acceptance condition of \mathcal{A}_\cap is inherited from \mathcal{A}_1 and $\overline{\mathcal{A}}_2$: a data ω -word σ over Σ is in $L(\mathcal{A}_\cap)$ iff \mathcal{A}_\cap has an infinite run $F_0 \xrightarrow{\sigma, 0} F_1 \xrightarrow{\sigma, 1} \dots$ where $F_0 = \{\langle q_I, [0] \sim \rangle\}$ and there exists i such that F_i contains only states in Q_1 . We then have that $L(\mathcal{A}_\cap) = L(\mathcal{A}_1) \cap L(\overline{\mathcal{A}}_2)$, so $L(\mathcal{A}_\cap)$ is empty iff $L(\mathcal{A}_1) \subseteq L(\mathcal{A}_2)$.

Let \mathcal{C}_\cap be the IPCANT computed from \mathcal{A}_\cap as in the proof of Theorem 3.2, except that the following step is added between steps (6) and (7), where q_\emptyset^2 is a new state and implementation is similar to that of step (5):

- (6 $\frac{1}{2}$) If $c(R) = 0$ for all R which intersect Q_2 , then pass through q_\emptyset^2 .

We thus have that $L(\mathcal{A}_\cap)$ is nonempty iff \mathcal{C}_\cap has an infinite run $\langle q_0, v_0 \rangle \xrightarrow{w_0, l_0} \langle q_1, v_1 \rangle \xrightarrow{w_1, l_1} \dots$ where $\langle q_0, v_0 \rangle$ is the initial configuration and there exists i such that $q_i = q_\emptyset^2$.

We define \preceq to be the following quasi-ordering on configurations of \mathcal{C}_\cap : $\langle q, v \rangle \preceq \langle q', v' \rangle$ iff $q = q'$ and $v \leq v'$. By Dickson's Lemma [Dickson 1913], \preceq is a *well-quasi-ordering*: for every infinite sequence s_0, s_1, \dots , there exist $i < j$ such that $s_i \preceq s_j$. Now, consider the following procedure:

- (i) Let S consist of the initial configuration of \mathcal{C}_\cap .
- (ii) Let S' be the set of all successors of configurations in S by lazy transitions.
- (iii) If for all $s' \in S'$ there exists $s \in S$ with $s \preceq s'$, then stop. Otherwise, set S to $S \cup S'$, and repeat from (ii).

Since \preceq is a well-quasi-ordering, the procedure terminates. Let S_{last} denote the value of S at the termination. It is a finite set, and by Proposition 2.7, its upward closure $\uparrow S_{\text{last}} = \{s' : \exists s \in S_{\text{last}} (s \preceq s')\}$ is the set of all configurations which \mathcal{C}_\cap can reach from the initial configuration.

To conclude decidability of inclusion for safety 1ARA₁, it remains to show that we can decide whether $\uparrow S_{\text{last}}$ contains a configuration whose state is q_\emptyset^2 and from which \mathcal{C}_\cap has an infinite run. But that is the case iff S_{last} contains such a configuration, and for any configuration $\langle q, v \rangle$, we have by the proof of Theorem 3.3 that \mathcal{C}_\cap has an infinite run from $\langle q, v \rangle$ iff it has a sequence of $m - 1$ lazy transitions from $\langle q, v \rangle$, where m is as computed in that proof.

We now turn to showing that already validity for safety $\text{LTL}_1^\downarrow(\mathbf{X}, \mathbf{R})$ is not primitive recursive. We reduce (in logarithmic space) from satisfiability over finite data words for $\text{LTL}_1^\downarrow(\mathbf{X}, \mathbf{F})$, which is not primitive recursive by [Demri and Lazić 2009, Theorem 5.2]. In negation normal form, the latter logic differs from safety $\text{LTL}_1^\downarrow(\mathbf{X}, \mathbf{R})$ by having temporal operators $\bar{\mathbf{X}}$, \mathbf{F} and \mathbf{G} instead of \mathbf{R} . Over finite data words, \mathbf{X} and its dual $\bar{\mathbf{X}}$ are distinct: at any final word position and for any ϕ , $\mathbf{X}\phi$ is false whereas $\bar{\mathbf{X}}\phi$ is true.

Consider the following translation from formulae of $\text{LTL}_1^\downarrow(\mathbf{X}, \mathbf{F})$ in negation normal form with alphabet Σ to formulae of co-safety $\text{LTL}_1^\downarrow(\mathbf{X}, \mathbf{R})$ with alphabet $\Sigma \uplus \{\times\}$. Only cases where the construct is modified are shown.

$$\begin{aligned} t(\mathbf{X}\phi) &= \mathbf{X}(t(\phi) \wedge \bigvee_{a \in \Sigma} a) & t(\mathbf{F}\phi) &= (\bigvee_{a \in \Sigma} a) \mathbf{U}(t(\phi) \wedge \bigvee_{a \in \Sigma} a) \\ t(\bar{\mathbf{X}}\phi) &= \mathbf{X}(t(\phi) \vee \times) & t(\mathbf{G}\phi) &= (t(\phi) \wedge \bigvee_{a \in \Sigma} a) \mathbf{U}\times \end{aligned}$$

Given a sentence ϕ , we have that a data ω -word σ over $\Sigma \uplus \{\times\}$ satisfies $\psi_\phi = t(\phi) \wedge (\bigvee_{a \in \Sigma} a) \wedge (\top \mathbf{U}\times)$ iff there exists $i > 0$ such that the i -prefix of σ does not contain \times and satisfies ϕ , and $\sigma(i) = \times$. It remains to observe that the dual of ψ_ϕ is a sentence of safety $\text{LTL}_1^\downarrow(\mathbf{X}, \mathbf{R})$, which is valid over data ω -words iff ϕ is satisfiable over finite data words. \square

6. CONCLUDING REMARKS

Satisfiability (over timed ω -words) for the safety fragment of metric temporal logic (MTL) was shown decidable in [Ouaknine and Worrell 2006], and nonelementary in [Bouyer et al. 2008] by reducing from termination of channel machines with emptiness testing and insertion errors. It would be interesting to investigate whether

ideas in the proof of Theorem 3.3 above can be combined with those in the proof of primitive recursiveness of termination of channel machines with occurrence testing and insertion errors [Bouyer et al. 2008] to obtain that satisfiability for safety MTL is primitive recursive.

Another open question is whether nonemptiness of safety forward alternating tree automata with 1 register [Jurdziński and Lazić 2007] is primitive recursive.

ACKNOWLEDGMENTS

I am grateful to Stéphane Demri and James Worrell for helpful discussions.

REFERENCES

- ALPERN, B. AND SCHNEIDER, F. B. 1987. Recognizing safety and liveness. *Distr. Comput.* 2, 3, 117–126.
- BJÖRKLUND, H. AND SCHWENTICK, T. 2007. On notions of regularity for data languages. In *Fundamentals of Comput. Theory (FCT), 16th Int. Symp.* Lect. Notes Comput. Sci., vol. 4639. Springer, 88–99.
- BOJAŃCZYK, M., DAVID, C., MUSCHOLL, A., SCHWENTICK, T., AND SEGOUFIN, L. 2006. Two-variable logic on data trees and XML reasoning. In *25th ACM SIGACT-SIGMOD-SIGART Symp. on Princ. of Database Systems (PODS)*. ACM, 10–19.
- BOJAŃCZYK, M., MUSCHOLL, A., SCHWENTICK, T., SEGOUFIN, L., AND DAVID, C. 2006. Two-variable logic on words with data. In *21th IEEE Symp. on Logic in Comput. Sci. (LICS)*. IEEE Comput. Soc., 7–16.
- BOUYER, P., MARKEY, N., OUAKNINE, J., SCHNOEBELEN, P., AND WORRELL, J. 2008. On termination for faulty channel machines. In *25th Int. Symp. on Theor. Asp. of Comput. Sci. (STACS)*. IBFI, Schloss Dagstuhl, Germany, 121–132.
- BRZozowski, J. A. AND LEISS, E. L. 1980. On equations for regular languages, finite automata, and sequential networks. *Theor. Comput. Sci.* 10, 1, 19–35.
- DAVID, C. 2004. Mots et données infinies. M.S. thesis, Laboratoire d’Informatique Algorithmique: Fondements et Applications, Paris.
- DEMRI, S. AND LAZIĆ, R. 2009. LTL with the freeze quantifier and register automata. *ACM Trans. On Comp. Logic* 10, 3.
- DICKSON, L. 1913. Finiteness of the odd perfect and primitive abundant numbers with distinct factors. *Amer. J. Math.* 35, 413–422.
- JURDZIŃSKI, M. AND LAZIĆ, R. 2007. Alternation-free modal mu-calculus for data trees. In *22nd IEEE Symp. on Logic in Comput. Sci. (LICS)*. IEEE Comput. Soc., 131–140.
- KAMINSKI, M. AND FRANCEZ, N. 1994. Finite-memory automata. *Theor. Comput. Sci.* 134, 2, 329–363.
- LAZIĆ, R. 2006. Safely freezing LTL. In *FSTTCS: Found. of Softw. Technology and Theor. Comput. Sci., 26th Int. Conf.* Lect. Notes Comput. Sci., vol. 4337. Springer, 381–392.
- LIPTON, R. J. 1976. The reachability problem requires exponential space. Tech. Rep. 62, Yale University.
- LÖDING, C. AND THOMAS, W. 2000. Alternating automata and logics over infinite words. In *Theor. Comput. Sci., Int. Conf. (IFIP TCS)*. Lect. Notes Comput. Sci., vol. 1878. Springer, 521–535.
- MULLER, D. E., SAOUDI, A., AND SCHUPP, P. E. 1986. Alternating automata, the weak monadic theory of the tree, and its complexity. In *Automata, Lang. and Program., 13th Int. Coll. (ICALP)*. Lect. Notes Comput. Sci., vol. 226. Springer, 275–283.
- NEVEN, F., SCHWENTICK, T., AND VIANU, V. 2004. Finite state machines for strings over infinite alphabets. *ACM Trans. On Comp. Logic* 5, 3, 403–435.
- OUAKNINE, J. AND WORRELL, J. 2006. Safety metric temporal logic is fully decidable. In *Tools and Algorithms for the Constr. and Anal. of Systems (TACAS), 12th Int. Conf.* Lect. Notes Comput. Sci., vol. 3920. Springer, 411–425.

- SAKAMOTO, H. AND IKEDA, D. 2000. Intractability of decision problems for finite-memory automata. *Theor. Comput. Sci.* 231, 2, 297–308.
- SEGOUFIN, L. 2006. Automata and logics for words and trees over an infinite alphabet. In *Comput. Sci. Logic (CSL), 20th Int. Works.* Lect. Notes Comput. Sci., vol. 4207. Springer, 41–57.
- VARDI, M. Y. 1996. An automata-theoretic approach to linear temporal logic. In *Banff Higher Order Works.* Lect. Notes Comput. Sci., vol. 1043. Springer, 238–266.

Received February 2008; revised March 2009; accepted April 2010